Modeling Syntactic-Semantic Dependency Correlations in Semantic Role Labeling Using Mixture Models

Junjie Chen*, **Xiangheng He**** and **Yusuke Miyao***
Department of Computer Science, The University of Tokyo, Tokyo*
GLAM – Group on Language, Audio, & Music, Imperial College London, UK**
{christopher, yusuke}@is.s.u-tokyo.ac.jp
x.he20@imperial.ac.uk

Abstract

In this paper, we propose a mixture model-based end-to-end method to model the syntactic-semantic dependency correlation in Semantic Role Labeling (SRL). Semantic dependencies in SRL are modeled as a distribution over semantic dependency labels conditioned on a predicate and an argument word. The semantic label distribution varies depending on Shortest Syntactic Dependency Path (SSDP) hop patterns. We target the variation of semantic label distributions using a mixture model, separately estimating semantic label distributions for different hop patterns and probabilistically clustering hop patterns with similar semantic label distributions. Experiments show that the proposed method successfully learns a cluster assignment reflecting the variation of semantic label distributions. Modeling the variation improves performance in predicting short distance semantic dependencies, in addition to the improvement on long distance semantic dependencies that previous syntax-aware methods have achieved. The proposed method achieves a small but statistically significant improvement over baseline methods in English, German, and Spanish and obtains competitive performance with state-of-the-art methods in English. [1]

1 Introduction

Semantic Role Labeling (SRL) answers an essential question about sentence semantics: “[Who] [does what] [to whom]”. A core problem of SRL is identifying semantic dependencies that specify the semantic role of arguments in relation to predicates (He et al., 2018; Kasai et al., 2019). For example, [who] (argument) is the agent (semantic role) to [does what] (predicate). Semantic dependency parsers (Dozat and Manning, 2018a) identify semantic dependencies by giving a distribution over semantic dependency labels (denoted as semantic label distribution) for all predicate-argument pairs.

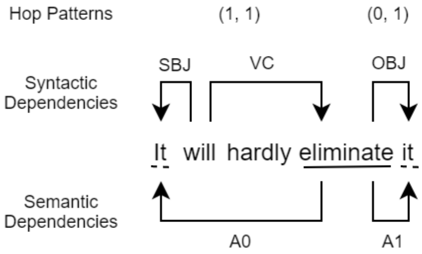


Figure 1: An example illustrating the impact of SSDPs on semantic label distributions. The solid underline highlights the predicate, and the dashed underline highlights the arguments.

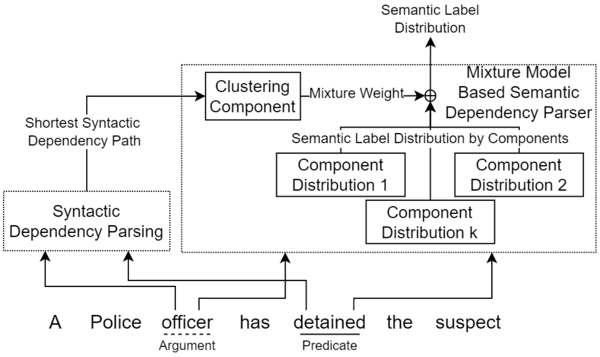

Semantic Label Distribution
Clustering Component
Mixture Weight
Mixture Model Based Semantic Dependency Parser
Shortest Syntactic Dependency Path
Semantic Label Distribution by Components
Component Distribution 1
Component Distribution 2
Component Distribution k
Syntactic Dependency Parsing
A Police officer has detained the suspect
Argument
Predicate

Figure 2: An illustration of the proposed mixture model-based method for semantic dependency parsing.

In this paper, we propose a mixture model (Pearson, 1894) based semantic dependency parser for SRL where we target the dependence of semantic label distributions on Shortest Syntactic Dependency Path (SSDP) patterns. SSDP is the shortest path connecting a predicate-argument pair in a syntactic dependency tree. Bunescu and Mooney (2005) and Cai et al. (2016) claim that SSDP encodes most information about bigram relations, such as the semantic dependency. Indeed, previous research (He et al., 2018; Xia et al., 2019) shows that modeling the correlation between SSDPs and semantic dependencies is crucial for building a high-performance SRL system.

[1] Our code is available at this repository.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 7959 - 7969
May 22-27, 2022 ©2022 Association for Computational Linguistics

Semantic label distributions vary depending on SSDPs, even when the SSDPs connect predicate-argument pairs with the same surface words. Figure 1 shows an example where two predicate-argument pairs have different semantic dependency labels while sharing the same surface words. SSDP patterns help discriminate semantic labels between the two pairs. The example indicates the dependence of semantic label distributions on SSDP patterns. We propose a mixture model-based method (Figure 2) to model the dependence in two steps: (1) Separately estimating semantic label distributions for different SSDP patterns as component distributions, and (2) Probabilistically clustering SSDP patterns with similar semantic label distributions using a mixture weight. The mixture model estimates the semantic label distribution by aggregating the component distributions using the mixture weight. We focus on *SSDP hop patterns* in this paper as we observed a drastic variation in semantic label distributions for different hop patterns through the change in mutual information (Shannon et al., 1949) (Section 2).

We evaluate the proposed method using the CoNLL-2009 dataset (Hajič et al., 2009) [2], the most popular multi-lingual SRL dataset with parallel syntactic and semantic dependency annotations. Experiments show that the proposed method correctly learns a mixture weight reflecting the variation in semantic label distributions. Modeling the variation improves performance in predicting short distance semantic dependencies in addition to long distance dependencies that previous syntax-aware methods (He et al., 2018; Roth and Lapata, 2016; Strubell et al., 2018) improve only on. Previous syntax-aware methods improve their performance on long distance dependencies at the expense of the performance on short distance dependencies. In comparison, the proposed method makes no such compromise, improving its performance over semantic dependencies of all ranges. In general, the proposed method obtains a small but statistically significant improvement over baseline methods in English, German, and Spanish and achieves competitive performance with state-of-the-art methods in English.

Our contributions are: (1) studying the variation

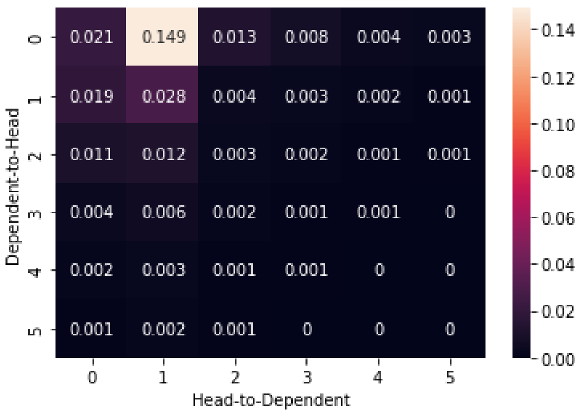


Figure 3: Mutual information gain of each SSDP hop pattern.

in semantic label distributions for different SSDP hop patterns, (2) proposing a mixture model-based method capturing the variation, and (3) conducting a detailed experiment evaluating the proposed method.

2 Motivation

As mentioned in Section 1, SSDP affects the choice of semantic dependency labels. We study the impact of SSDP hop patterns on semantic label distributions through the change in mutual information (Shannon et al., 1949) in this section. We observe a drastic change in mutual information only for hop patterns that frequently co-occur with semantic dependencies.

SSDP is the path connecting a predicate-argument pair in a syntactic dependency tree. Its hop pattern describes the number of transitions needed to transit from the predicate to the argument. We denote the hop pattern by (α, β), where α is the number of dependent-to-head transitions and β is the number of head-to-dependent transitions. In a syntactic dependency tree, syntactic dependencies are arcs pointing from syntactic heads to syntactic dependents. The head-to-dependent transition moves in the same direction as the syntactic dependencies, whereas the dependent-to-head transition moves in the opposite direction. In Figure 1, the SSDP connecting "eliminate" and "it" consists of a dependent-to-head transition moving from "eliminate" to "will", and a head-to-dependent transition moving from "will" to "it". The hop pattern of this SSDP is (1, 1).

We denote the syntactic random variable for hop patterns as X and the semantic random variable for semantic labels as Y. X maps predicate-argument

[2] While being similar, the CoNLL-2009 dependency format (Surdeanu et al., 2008) predates the well-known universal dependency format (Nivre et al., 2017) and have fundamental differences. This work follows the CoNLL-2009 format for both syntactic and semantic dependencies.

word pairs (p^s, a^s) in a sentence s to their hop patterns, whereas Y maps the pairs to their semantic labels. Their mutual information $\mathrm{MI}(X, Y)$ measures the reduction in uncertainty about Y after knowing X. High mutual information indicates relatively low uncertainty in the conditional distribution $P_{Y|X}$.

To highlight the impact of hop patterns on semantic label distributions, we compare the mutual information of two ideal models, a syntax-aware model $(X_{(\alpha,\beta)}, Y)$ and a syntax-agnostic model (X_0, Y). We define the syntactic variables $X_{(\alpha,\beta)}$ and X_0 as Equation 1 and 2. This definition makes the variable $X_{(\alpha,\beta)}$ sensitive only to the hop pattern (α, β) and X_0 blind to any hop pattern information. We define the mutual information gain of (α, β) as the difference in mutual information between the syntax-aware model and the syntax-agnostic model (Equation 3).

$$X_{(\alpha,\beta)}(p^s, a^s) = \begin{cases} 1, (p^s, a^s) \text{ is of } (\alpha, \beta) \\ 0, \text{otherwise} \end{cases} \tag{1}$$

$$X_0(p^s, a^s) = 0 \tag{2}$$

$$\Delta\mathrm{MI}(X_{(\alpha,\beta)}, X_0) = \mathrm{MI}(X_{(\alpha,\beta)}, Y) - \mathrm{MI}(X_0, Y) \tag{3}$$

Figure 3 reports the mutual information gain of each hop pattern using the English training set of the CoNLL-2009 dataset. The figure shows that different hop patterns have drastically varying mutual information gains. A sharp spike of mutual information gain occurs in the hop pattern (0, 1) with a gain value of 0.149 bits, indicating a strong impact of the hop pattern (0, 1) on semantic label distributions. Hop patterns with relatively short transitions have non-zero gains ranging from 0.011 bits to 0.149 bits, which indicates the degree of impact differs drastically. These hop patterns frequently co-occur with semantic dependencies (He et al., 2018). On the other hand, hop patterns co-occurring rarely with semantic dependencies have long transitions. These hop patterns have near-zero mutual information gains in Figure 3, which indicates the weak impact of the patterns. The varying degree of impact motivates the separate estimation of semantic label distributions for different hop patterns. The amount of hop patterns with a weak impact motivates the clustering of hop patterns that share similar semantic label distributions.

3 Background

In this section, we present background information about syntactic and semantic dependency parsing and mixture models. We also present a brief survey about syntax-aware SRL methods using SSDP information and compare the proposed method with the previous methods.

3.1 Syntactic and Semantic Dependency Parsing

Both syntactic and semantic dependencies describe bigram relations between words, namely heads and dependents. The heads and the dependents correspond to syntactic heads and dependents in syntactic dependencies and predicates and arguments in semantic dependencies. The similarity suggests that a mechanism, such as the biaffine parser (Dozat and Manning, 2017, 2018b), can capture the two dependencies. For semantic dependencies, the biaffine parser estimates a distribution $P(r|p, a)$ over relations $r \in \mathcal{R} \bigcup \{\epsilon\}$ between a predicate p and an argument a. $\mathcal{R}$ denotes the set of semantic relation labels, and ϵ denotes no relation occurring between p and a. For syntactic dependencies, the biaffine parser estimates a distribution $P(h|d)$, predicting the syntactic head h of the syntactic dependent d. Neural biaffine parsers estimate the two distributions as Equation 4[3], 5, 6[4], and 7. e_p, e_a, e_h and e_d denote the feature vectors of p, a, h and d from a sentence encoder.

$$\phi^r(e_p, e_a) = e_p^T(W_1^r)e_a + w_2^r([e_p; e_a]) + b^r \tag{4}$$

$$P(r|p, a) = \mathrm{Softmax}([\phi^r(e_p, e_a)]) \tag{5}$$

$$\phi(e_h, e_d) = e_h^T(W_1)e_d + w_2([e_h; e_d]) + b \tag{6}$$

$$P(h|d) = \mathrm{Softmax}([\phi(e_h, e_d)]) \tag{7}$$

3.2 Mixture Model and Latent Variable Model

Mixture models assume data to be generated from a mixture distribution whose component distributions belong to the same distributional family, such as the Gaussian distributions, but possess distinct parameters. The mixture of component distributions grants additional flexibility to the mixture model. For example, the Gaussian mixture model can capture multi-mode phenomena as opposed to the simple Gaussian model (Bishop and Nasrabadi, 2007). A mixture model contains two core variables: an observable data variable

[3] W_1^r is a weight matrix, w_2^r is a weight vector, and b^r is a bias term for estimating the unnormalized probability score of $P(r|p, a)$.

[4] Similarly, W_1, w_2, and b are parameters for estimating the unnormalized score of $P(h|d)$.

X and a latent variable C indexing the component distribution that generates the data. The mixture model computes the marginal likelihood $P_\theta(x) := \sum_c P_\theta(x|c)P_\theta(c)$ by aggregating its component distributions $P_\theta(x|c)$ using the mixture weight $P_\theta(c)$. The optimal parameter (i.e., the mixture weight and the parameters of component distributions) can be estimated by maximizing the log-likelihood $\log P_\theta(x)$. However, direct maximum likelihood estimation on the marginal log-likelihood is intractable for mixture models (Murphy, 2012), and the conventional Expectation-Maximization algorithm (Dempster et al., 1977) requires finding optimal parameters at each iteration. Variational Inference (Xu et al., 2015; Ba et al., 2015) maximizes a variational lowerbound of the log-likelihood (Equation 8), simultaneously optimizing the component distributions and the mixture weight.

$$\mathcal{L} = \sum_c q(c|x) \log \frac{P_\theta(x|c)P_\theta(c)}{q(c|x)} \tag{8}$$

$$= \log P_\theta(x) - \mathrm{KL}(q(c|x)||P_\theta(c|x)) \tag{9}$$

3.3 Syntactic Dependency Information in Semantic Dependency Parsing

Inspired by the close connection of syntactic and semantic dependencies, He et al. (2018), Roth and Lapata (2016), and Shi et al. (2020) attempt to build high-performance SRL systems using SSDP information. While the research improves performance over syntax-agnostic methods, their methods either require language-specific hyperparameters or exhibit a behavior challenging to interpret.

The pruning method (He et al., 2018, 2019) is readily interpretable but requires language-specific hyperparameters. The method utilizes a statistical bias that most SSDPs rarely co-occur with semantic dependencies. It eliminates predicate-argument pairs of the infrequent SSDPs using heuristics. Whether an SSDP can co-occur with semantic dependencies is hardcoded in heuristics, making the method highly interpretable. However, the heuristics are language-specific, requiring manual tuning for every language.

The neural methods (Roth and Lapata, 2016; Foland and Martin, 2015) are more language-independent but suffer from limited interpretability. The methods implicitly encode SSDP information using neural network encoders. Roth and Lapata (2016) and Foland and Martin (2015) encode SSDPs in a continuous embedding using an Long-Short Term Memory (LSTM) model or a Convolutional Neural Network model. Shi et al. (2020) jointly learns SSDP and semantic dependency information using a Transformer (Vaswani et al., 2017) by merging SSDP information with semantic dependency labels. The research reports performance improvements in one or more languages. However, interpreting the model's behavior is challenging. Neural encoders, such as the LSTM model in Roth and Lapata (2016), project SSDPs in a high-dimensional space. The high-dimensional space has a complex structure, rendering clustering analyses based on Euclidean distances less effective. Roth and Lapata (2016) interprets the behavior of their model using the clustering analysis, suggesting that their model captures many linguistic phenomena. However, the linguistic phenomena are fragmental and limited to a few syntactic constructions.

In contrast, the proposed method is generic like the neural methods and interpretable like the pruning method. The proposed method optimizes its parameters using gradients of the back-propagated errors, which makes the proposed method more language-independent. As a result, the proposed method learns a mixture weight reflecting the impact of SSDP hop patterns on semantic label distributions, enabling analyses using the mixture weight.

4 Proposal

In this section, we present the proposed mixture model-based semantic dependency parser to model the dependence of semantic label distributions on SSDP hop patterns. In Section 2, we discussed the need to separately estimate semantic label distributions for different hop patterns and the need to cluster hop patterns sharing similar semantic label distributions. The proposed parser estimates semantic label distributions for different hop patterns using the component distributions and clusters hop patterns using the mixture weight of a mixture model.

Figure 2 illustrates the model architecture of the proposed method. The model contains a conventional biaffine parser for syntactic dependencies and a mixture model-based biaffine parser for semantic dependencies. The syntactic parser provides a syntactic dependency tree from which the clustering component extracts hop patterns and determines the mixture weights. The biaffine parsers in the semantic parser estimate the component distributions. The semantic parser computes the semantic label distribution by aggregating the component distributions using the mixture weight. The syntactic and the semantic parser share a backbone sentence encoder, a Transformer model in our implementation. We jointly optimize the parameters of the syntactic and the semantic parser by optimizing the log-likelihood of the syntactic dependencies and a variational lowerbound of the log-likelihood (ELBo) of the semantic dependencies. We use the lowerbound as an approximation to the log-likelihood for inference because we find it works best in predicting semantic dependencies.

We expand on the training objective of the semantic parser. The objective is to maximize the likelihood $P_\theta(r|p,a)$ of the observed semantic label r conditioned on the predicate p and the argument a. We rewrite the likelihood as a marginal of the joint likelihood $P_\theta(r,c|p,a)$ where c is the index of the component distributions. The joint likelihood can be decomposed as Equation 12 where the former term corresponds to the component distributions and the latter term corresponds to the mixture weight. Since we are interested in separating semantic label distributions by hop patterns, we replace the term $P_\theta(c|p,a)$ with $P_\theta(c|\mathrm{ssdp}(p,a))$ where $\mathrm{ssdp}(p,a)$ maps predicate-argument pairs to their hop patterns. $P_\theta(c|\mathrm{ssdp}(p,a))$ also serves as the variational approximation $q(c|r,p,a)$ because we assume the hop pattern, together with the predicate-argument pair, determines the semantic dependency label. This assumption removes the need to condition the component index c on the semantic label r in the variational approximation q. In this implementation, we encode hop patterns with orthogonally initialized embeddings and estimate the mixture weight of a hop pattern by applying a multi-layer perceptron followed by a softmax layer to the embedding.

$$
\begin{aligned}
&\log P_\theta(r|p,a) && (10)\\
&\quad = \log \sum_c P_\theta(r,c|p,a) && (11)\\
&\quad = \log \sum_c P_\theta(r|c,p,a) P_\theta(c|p,a) && (12)\\
&\quad = \log \sum_c P_\theta(r|c,p,a) P_\theta(c|\mathrm{ssdp}(p,a)) && (13)\\
&\quad \geq \sum_c P_\theta(c|\mathrm{ssdp}(p,a)) \log P_\theta(r|c,p,a) && (14)\\
&\quad = \mathcal{L}_{sem}(\theta|\mathcal{X}_{sem} = (r,p,a)) && (15)
\end{aligned}
$$

Equation 16 depicts the full objective of the proposed model. It consists of a log-likelihood objective of the syntactic parser (Equation 17) and the ELBo objective of the semantic parser (Equation 15). $\mathcal{G}_{syn}$ stands for the set of all syntactic dependencies (h,d), whereas $\mathcal{G}_{sem}$ stands for the set of all semantic dependencies (r,p,a).

$$
\begin{aligned}
\mathcal{J}(\theta) = &\sum_{(h,d)\in\mathcal{G}_{syn}} \mathcal{L}_{syn}(\theta|\mathcal{X}_{syn} = (h,d))\\
&+ \sum_{(r,p,a)\in\mathcal{G}_{sem}} \mathcal{L}_{sem}(\theta|\mathcal{X}_{sem} = (r,p,a))
\end{aligned} \tag{16}
$$

$$
\mathcal{L}_{syn}(\theta|\mathcal{X}_{syn} = (h,d)) = \log P_\theta(h|d) \tag{17}
$$

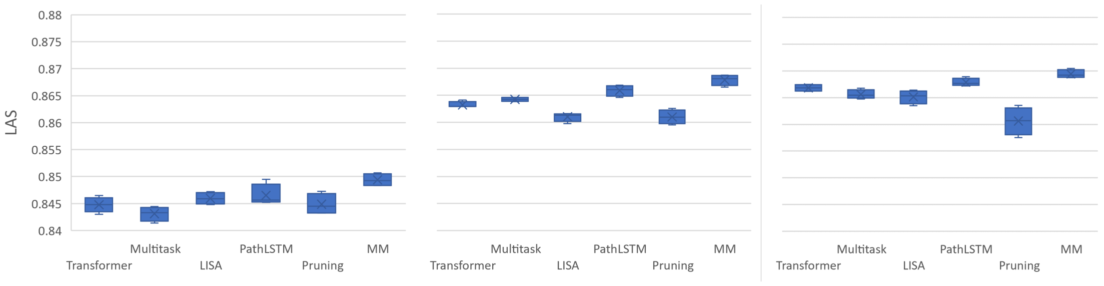
5 Experiment

In this section, we present experimental results for the proposed method. We call the proposed method as MM (mixture-model) in this section. We use the labeled attachment score (LAS) (Hajič et al., 2009) as the primary metric. LAS is a micro-F1 score measuring how well a model recovers semantic dependencies. We conduct our experiments comparing MM with five baseline methods (Table 1) using the CoNLL-2009 dataset. We perform the comparison on all languages using the corresponding development sets. Each model will run using four randomly generated seeds to mitigate the impact of the seeds. We also compare the semantic scores (Hajič et al., 2009) of MM with state-of-the-art syntax-aware methods using the English test set. The semantic score is a micro-F1 score evaluating models' performance in the predicate identification in addition to the semantic dependency recovery. We use preidentified predicates extracted from the mate-tools (Björkelund et al., 2010), following the evaluation method of Roth and Lapata (2016).

We evaluate MM using three word embeddings: a non-context-sensitive embedding, FastText (Joulin et al., 2016), and two context-sensitive embeddings, ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019). When comparing with state-of-the-art methods, we report results on the GloVe (Pennington et al., 2014) and the FastText embedding. However, the result on the FastText embedding is for reference only because the state-of-the-art methods report results mainly on the GloVe embedding. We use an 8-layer Transformer as the backbone encoder for MM and baseline models. We set the batch size to 5000 words, the maximum size that a P100 device can accommodate. We use the Adam optimizer (Kingma and Ba, 2015) with

Method Name	Syn	Description
Transformer	No	A Transformer model (Vaswani et al., 2017) using a biaffine semantic dependency parser
Multitask	Yes	A Transformer model using two biaffine parsers for syntactic and semantic dependencies
LISA	Yes	The Linguistically-Informed Self-Attention model (Strubell et al., 2018)
PathLSTM	Yes	A Multitask model using dependency path embeddings (Roth and Lapata, 2016)
Pruning	Yes	A Multitask model using the pruning technique (He et al., 2018)
MM	Yes	A Multitask model using the proposed mixture model-based semantic dependency parser

Table 1: Descriptions of baseline methods. Syn indicates whether a method is syntax-aware.



(a) FastText

(b) ELMo

(c) BERT

Figure 4: LAS of MM and baseline methods on the English development set.

	FastText	ELMo	BERT
Transformer	0.003	0.001	0.002
Multitask	0.000	0.003	0.001
LISA	0.005	0.000	0.002
PathLSTM	*0.058*	0.032	0.025
Pruning	0.011	0.000	0.005

Table 2: P-values of the significance tests. Each cell shows the p-value of a test comparing MM with a baseline method (shown in the row) on an embedding (shown in the column). P-values higher than the 0.05 threshold are highlighted in italic.

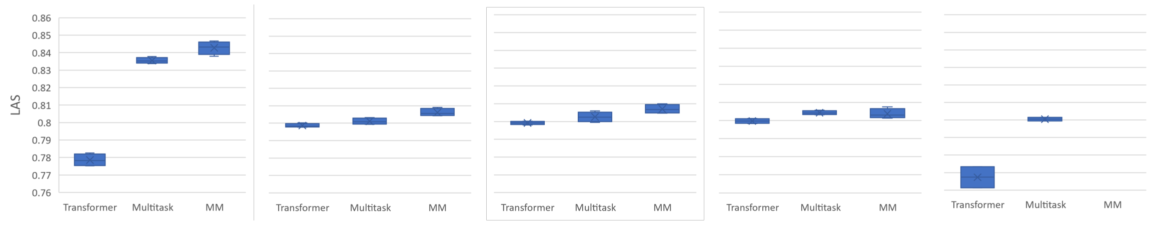
parameters $lr = 4e^{-6}$, $\beta_1 = 0.9$, and $\beta_2 = 0.98$ for training.

We set the number of component distributions k in MM to 5 for all languages. We find that this number works for most languages in a preliminary experiment exploring $k = 1, 3, 5, 7, 10$. For $k > 5$, some components will not be assigned to any hop pattern, resulting in a waste of model parameters. For $k < 5$, some components are forced to estimate semantic label distributions for hop patterns of different nature, resulting in a loss of performance. We do not perform back-propagation between the syntactic and the semantic parser in MM because we found the back-propagation causes negative impacts on the two parsers.

5.1 Comparison with Baselines

We find that MM significantly improves over baseline methods on the English development set. Figure 4 reports the LAS of MM and baseline methods using box plots. MM achieves better LAS than baseline methods in all three embeddings. We conduct a series of significance tests against a null hypothesis that MM performs equally to each baseline method. The p-values of the hypothesis tests are shown in Table 2. Each cell in the table shows the p-value of a test comparing MM with a baseline method (shown in the row) on an embedding (shown in the column). The table suggests that MM significantly outperforms all baseline methods on the three embeddings, except the PathLSTM method on the FastText embedding. The significance test confirms the effectiveness of MM in modeling semantic dependencies.

We find that MM learns a mixture weight reflecting the impact of hop patterns on semantic label distributions. Table 3 reports the component assignment extracted from the learned mixture weight. We extract the component assignment for hop patterns up to (5, 3). Most evidently, MM consistently assigns the hop pattern (0, 1) to a unique component in all three embeddings. This behavior agrees with our findings in Section 2 that the hop pattern has the highest mutual information gain. MM also consistently assigns hop patterns with near-zero mutual information gains to a single component. Moreover, MM clusters hop patterns with similar non-zero gains to a single component. These results suggest that semantic label distributions of different hop patterns have unique properties.



(a) German (b) Spanish (c) Catalan (d) Chinese (e) Czech†

Figure 5: LAS of MM, the Transformer, and the Multitask method on the development sets of German, Spanish, Catalan, Chinese, and Czech. The methods are trained on the FastText embedding. The Y-axis shows the LAS of each method. In Czech†, MM has an average LAS score of 0.4 and, therefore, can not be plotted in the figure.

↓ ↑	0	1	2	3
0	3	1	0	0
1	2	2	0	0
2	2	2	0	0
3	2	0	0	0
4	0	0	0	0
5	0	0	0	0

(a) FastText

↓ ↑	0	1	2	3
0	2	0	4	4
1	2	1	4	4
2	3	3	4	4
3	4	4	4	4
4	4	4	4	4
5	4	4	4	4

(b) ELMo

↓ ↑	0	1	2	3
0	4	0	2	2
1	4	3	2	2
2	4	3	2	2
3	4	4	2	2
4	2	2	2	2
5	2	2	2	2

(c) BERT

Table 3: Component assignments extracted from the mixture weight learned using the English dataset. Numbers headed with ↑ index dependent-to-head transitions, and numbers headed with ↓ index head-to-dependent transitions.

MM is readily applicable for other languages beyond English using the same hyperparameter setting. Figure 5 reports the comparison of MM with the Transformer and the Multitask method on the development sets of German, Spanish, Catalan, Chinese, and Czech. The Multitask method consistently outperforms the Transformer method in all languages. In comparison, MM significantly outperforms the Multitask method in German and Spanish. MM also has an arguable improvement over the Multitask method in Catalan. In Chinese, MM performs similarly to the Multitask method but better than the Transformer method. In Czech, MM somehow fails to learn and achieves a considerably low LAS. We might need to tune the architecture or hyperparameters here, while MM stably outperforms the baseline methods in other languages.

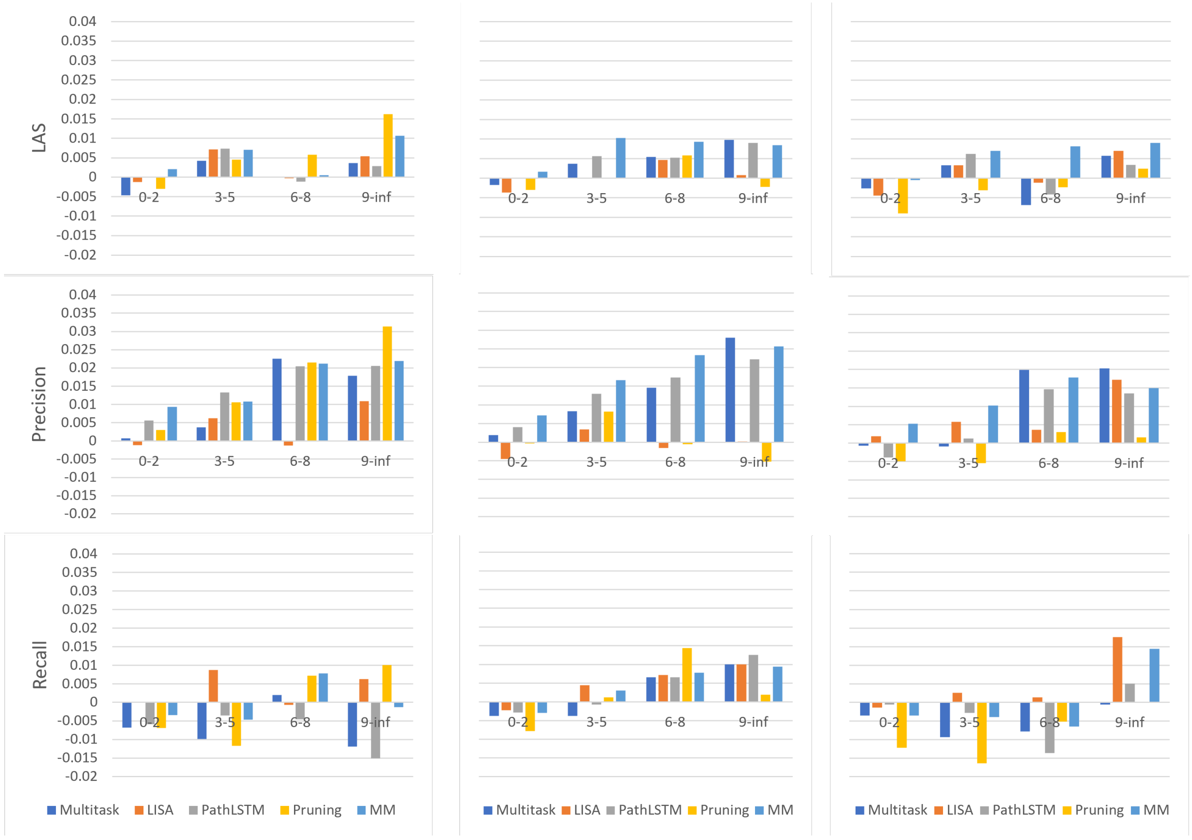
	WSJ			Brown		
GloVe	P	R	F1	P	R	F1
Zhou et al. (2020)	88.73	89.83	89.28	82.46	83.2	**82.82**
Li et al. (2019)	87.8	88	87.9	77	76.8	76.9
He et al. (2018)	89.7	89.3	89.5	81.9	76.9	79.3
Roth and Lapata (2016)	88.1	85.3	86.7	76.9	73.8	75.3
Kasai et al. (2019)	89	88.2	88.6	78	77.2	77.6
MM	91.03	90.13	**90.58**	80.59	79.21	79.83
MM (FastText)	91.16	90.19	90.71	83.93	82.64	83.28
ELMo	P	R	F1	P	R	F1
Li et al. (2019)	90.5	92.1	91.3	81.7	81.9	81.8
Kasai et al. (2019)	90.3	90	90.2	81	80.5	80.8
Cai and Lapata (2019)	91.7	90.8	91.2	83.2	81.9	82.5
Lyu et al. (2019)	-	-	91	-	-	82.2
Chen et al. (2019)	-	-	91.1	-	-	82.7
MM	92.21	91.45	**91.82**	86.51	85.30	**85.90**
BERT	P	R	F1	P	R	F1
Shi and Lin (2019)(base)	92.1	91.9	92	85.6	84.7	85.1
Shi and Lin (2019)(large)	92.4	92.3	**92.4**	85.7	85.8	85.7
Zhou et al. (2020)	91.21	91.19	91.2	85.65	86.09	85.87
MM	92.33	91.77	92.05	87.00	85.98	**86.32**

Table 4: Semantic scores of MM and state-of-the-art methods on the English test set. P, R, F1 stands for Precision, Recall, and F1 scores. Methods with the best performance score are highlighted in bold.

Using the Transformer method as a baseline, we find MM improves on both short and long distance semantic dependencies, whereas syntax-aware baseline methods improve only on long distance dependencies. To illustrate the finding, we group the semantic dependencies by their linear length[5] and evaluate the methods' performance on each group. We group the semantic dependencies into four bins: the short-distance bin (0-2) and the long-distance bins (3-5, 6-8, 9-inf). We then compute the relative performance score[6] of each syntax-aware method using the model with the median LAS score. Figure 6 reports the relative scores of LAS, Precision, and Recall. MM has the best relative LAS among syntax-aware methods in predict-

[5] $l = |\mathrm{idx}_p - \mathrm{idx}_a|$, where l is the linear length of the SSDP connecting the predicate p and the argument a. idx_p and idx_a represents the index of the predicate and the argument in the sentence.

[6] $s_r = s_{syn+} - s_{syn-}$, where s_r represents the relative score, s_{syn+} represents the score of the syntax-aware model, and s_{syn-} represents the score of the syntax-agnostic Transformer model.



(a) FastText

(b) ELMo

(c) BERT

Figure 6: Relative LAS, Precision, and Recall of syntax-aware models on the English development set. The X-axis shows the linear distance of semantic dependencies that each bin contains, the Y-axis shows the relative score.

ing short distance dependencies. On the FastText and the ELMo embedding, MM is the only method scoring a positive relative LAS (i.e., MM is the only method improving over the Transformer method). The reason is that MM achieves significantly better precision than baseline syntax-aware methods, which allows MM to overcome the lower recall. Meanwhile, MM has a performance improvement similar to the baseline syntax-aware methods in predicting long distance dependencies.

5.2 Comparison with the State-of-the-arts

MM achieves competitive performance with state-of-the-art syntax-aware methods. Table 4 reports the median semantic scores of MM and the reported scores of state-of-the-art methods on the English test set. The test set contains two sections: WSJ (in-domain) section and Brown (out-of-domain) section. MM achieves the best performance on the WSJ section on the GloVe and the ELMo embedding and performs comparably to other methods on the BERT embedding. MM also scores the best performance on the Brown section on the ELMo and the BERT embedding. We also find that MM on the FastText embedding performs better than MM on the GloVe embedding. This result is in line with a study evaluating non-context-sensitive word embeddings (Wang et al., 2019) where the FastText embedding outperforms the GloVe embedding on downstream NLP tasks.

6 Conclusion

This paper presented a mixture model-based method for syntax-aware semantic dependency parsing in SRL. The method models the dependence of semantic label distributions on SSDP patterns. We focused on SSDP hop patterns because we observed a drastic variation in semantic label distributions through the change in mutual information. The proposed method successfully learned a mixture weight reflecting the variation. The method improved performance in predicting both short and long distance semantic dependencies, whereas baseline syntax-aware methods improved only on long distance dependencies. The method outperformed baseline methods by a small but statistically significant margin in many languages. Moreover, the proposed method achieved performance competitive with state-of-the-art methods in English. Nonetheless, hop patterns contain only limited information about SSDP. In the future, we plan to apply the proposed method to more informative SSDP patterns, such as labeled SSDP patterns.

7 Acknowledgement

This research was supported by JST, CREST Grant Number JPMJCR2114, Japan.

References

Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. 2015. Multiple object recognition with visual attention. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Christopher M. Bishop and Nasser M. Nasrabadi. 2007. Pattern recognition and machine learning. *J. Electronic Imaging*, 16:049901.

Anders Björkelund, Bernd Bohnet, Love Hafdell, and Pierre Nugues. 2010. A high-performance syntactic and semantic dependency parser. In *Coling 2010: Demonstrations*, pages 33–36, Beijing, China. Coling 2010 Organizing Committee.

Razvan Bunescu and Raymond Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 724–731, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Rui Cai and Mirella Lapata. 2019. Semi-supervised semantic role labeling with cross-view training. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1018–1027, Hong Kong, China. Association for Computational Linguistics.

Rui Cai, Xiaodong Zhang, and Houfeng Wang. 2016. Bidirectional recurrent convolutional neural network for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 756–765, Berlin, Germany. Association for Computational Linguistics.

Xinchi Chen, Chunchuan Lyu, and Ivan Titov. 2019. Capturing argument interaction in semantic role labeling with capsule networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5415–5425, Hong Kong, China. Association for Computational Linguistics.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the *EM* algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Timothy Dozat and Christopher D. Manning. 2018a. Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2018b. Simpler but More Accurate Semantic Dependency Parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia. Association for Computational Linguistics.

William Foland and James Martin. 2015. Dependency-based semantic role labeling using convolutional neural networks. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 279–288, Denver, Colorado. Association for Computational Linguistics.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 1–18, Boulder, Colorado. Association for Computational Linguistics.

Shexia He, Zuchao Li, and Hai Zhao. 2019. Syntax-aware multilingual semantic role labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5350–5359, Hong Kong, China. Association for Computational Linguistics.

Shexia He, Zuchao Li, Hai Zhao, and Hongxiao Bai. 2018. Syntax for semantic role labeling, to be, or not to be. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2061–2071, Melbourne, Australia. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hervé Jégou, and Tomás Mikolov. 2016. Fasttext.zip: Compressing text classification models. *CoRR*, abs/1612.03651.

Jungo Kasai, Dan Friedman, Robert Frank, Dragomir Radev, and Owen Rambow. 2019. Syntax-aware neural semantic role labeling with supertags. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 701–709, Minneapolis, Minnesota. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Zuchao Li, Shexia He, Junru Zhou, Hai Zhao, Kevin Parnow, and Rui Wang. 2019. Dependency and span, cross-style semantic role labeling on propbank and nombank. *CoRR*, abs/1911.02851.

Chunchuan Lyu, Shay B. Cohen, and Ivan Titov. 2019. Semantic role labeling with iterative structure refinement. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1071–1082, Hong Kong, China. Association for Computational Linguistics.

Kevin P. Murphy. 2012. *Machine learning - a probabilistic perspective*. Adaptive computation and machine learning series. MIT Press.

Joakim Nivre, Daniel Zeman, Filip Ginter, and Francis Tyers. 2017. Universal Dependencies. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts*, Valencia, Spain. Association for Computational Linguistics.

Karl Pearson. 1894. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society A*, 185:71–110.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Michael Roth and Mirella Lapata. 2016. Neural semantic role labeling with dependency path embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1192–1202, Berlin, Germany. Association for Computational Linguistics.

Claude E. Shannon, Warren Weaver, and Norbert Wiener. 1949. The mathematical theory of communication. *Physics Today*, 3:31–32.

Peng Shi and Jimmy Lin. 2019. Simple BERT models for relation extraction and semantic role labeling. *CoRR*, abs/1904.05255.

Tianze Shi, Igor Malioutov, and Ozan Irsoy. 2020. Semantic role labeling as syntactic dependency parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7551–7571, Online. Association for Computational Linguistics.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium. Association for Computational Linguistics.

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning - CoNLL '08*, page 159, Manchester, United Kingdom. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Bin Wang, Angela Wang, Fenxiao Chen, Yun Cheng Wang, and C.-C. Jay Kuo. 2019. Evaluating word embedding models: methods and experimental results. *APSIPA Transactions on Signal and Information Processing*, 8.

Qingrong Xia, Zhenghua Li, Min Zhang, Meishan Zhang, Guohong Fu, Rui Wang, and Luo Si. 2019. Syntax-aware neural semantic role labeling. In *The*

Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, pages 7305–7313. AAAI Press.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2048–2057. JMLR.org.

Junru Zhou, Zuchao Li, and Hai Zhao. 2020. Parsing all: Syntax and semantics, dependencies and spans. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4438–4449, Online. Association for Computational Linguistics.